Efficient and principled Method for detecting Communities in Networks

Brian Ball, Brian Karrer, and M. E. J. Newman

Physical Review, 2011

Outline

The concept of "community"

- There is no generally accepted definition of what a community is.
- Communities in a network are groups of vertices:
  - with relatively dense connections within groups
  - but sparser connections between them.
- **Note:** They may be disjoint or overlapping.

Detecting community

- Detecting communities is a fundamental problem.

The paper's purpose

- The authors derive an algorithm for community detection applied to overlapping and non overlapping communities.
- It gives good results on both real-world networks and synthetic benchmarks, competitive with previous methods.

An Illustration

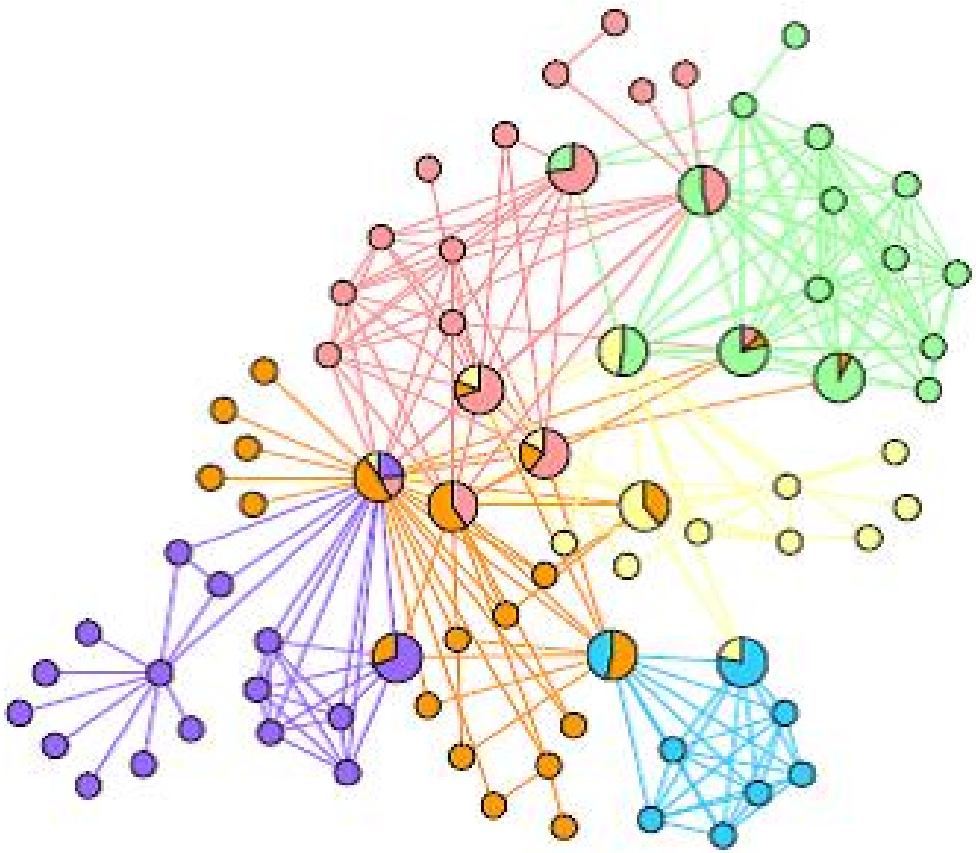

Figure: An illustration for a result of the algorithm.

A Generative Network Model

Definition 1

- *with a given number n of vertices*
- *with a given number K of communities*
- *has $n \cdot K$ parameters θ_{iz} representing the propensity of vertex i to have edges of color z.*
- $$E_{ijz} \sim \begin{cases} Pois(\theta_{iz} \cdot \theta_{jz}) & if \quad i \neq j \\ Pois(\frac{1}{2} \cdot \theta_{iz} \cdot \theta_{jz}) & if \quad i = j \end{cases} \tag{1}$$

Detecting Overlapping Communities

- Given a undirected graph G with $n \times n$ adjacency matrix $A, (A_{ij} = 0, 1 or 2)$
- We fit the model to G by maximizing $\log P(G \mid \theta)$

$$\log P(G \mid \theta) = \sum_{ij} A_{ij} \cdot \log(\sum_{z} \theta_{iz} \cdot \theta_{jz}) - \sum_{ijz} \theta_{iz} \cdot \theta_{jz}$$

$$\geq \sum_{ijz} [A_{ij} \cdot q_{ij}(z) \cdot \log \frac{\theta_{iz} \cdot \theta_{jz}}{q_{ij}(z)} - \theta_{iz} \cdot \theta_{jz}] \quad (2)$$

for all $q_{ij}(z) > 0$ satisfying $\sum_{z} q_{ij}(z) = 1$.

Detecting Overlapping Communities

- Maximizing the log likelihood is to solve following equations:

$$q_{ij}(z) = \frac{\theta_{iz} \cdot \theta_{jz}}{\sum_z \theta_{iz} \cdot \theta_{jz}} \tag{3}$$

$$\theta_{iz} = \frac{\sum_j A_{ij} \cdot q_{ij}(z)}{\sqrt{\sum_{ij} A_{ij} \cdot q_{ij}(z)}} \tag{4}$$

  - choose a random set of initial values.
  - iterate until $\theta_{iz}^{(l)}$ converge.
- This is Expectation-Maximization (EM) algorithm

Output

- If $q_{ij}(z_0) = \max_z q_{ij}(z)$ then the edge connecting i and j belongs to community z_0.
- One vertex can belong to more than one community.

Implementation

- To save memory use, we calculate number of edges of color z connected to i:

$$k_{iz} = \sum_j A_{ij} \cdot q_{ij}(z). \tag{5}$$

- Firstly, initialize $\{k_{iz}\}$ then compute new values $\{k_{iz}\}$ until $\{k_{iz}\}$ converge.
- Since most of them converge to 0, hence the authors give 2 strategies for pruning set of variables by a certain threshold δ.

Test on synthetic networks

- Test on synthetic networks with known communities and having similar properties.
- Calculate the fractions of vertices assigned to the correct community.
- Calculate the Jaccard index.

Test on real-world networks

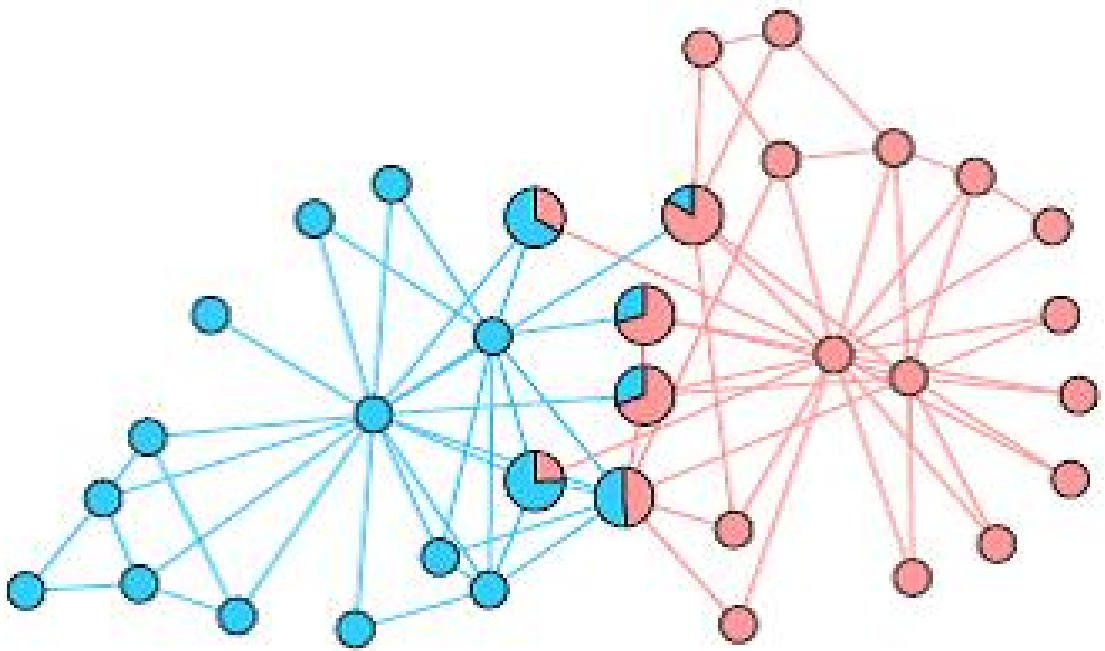

Figure: Karate club network.

Test on real-world networks

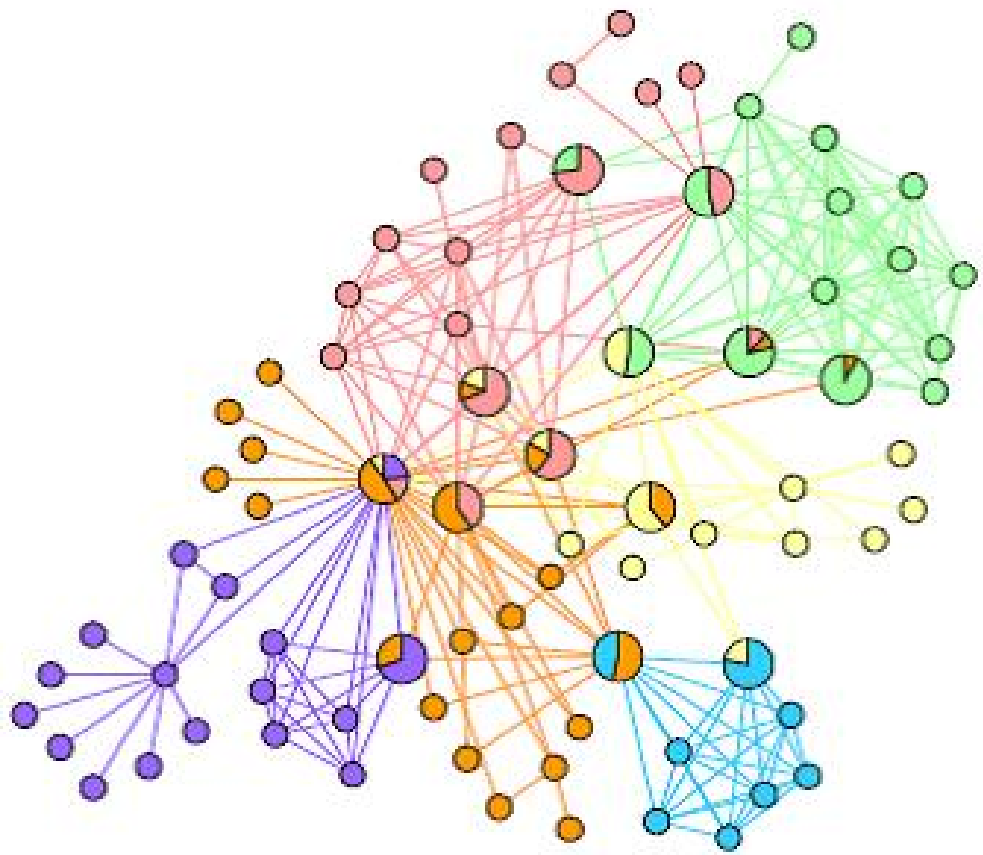

Figure: The network of characters from Les Misérables.

Test on real-world networks

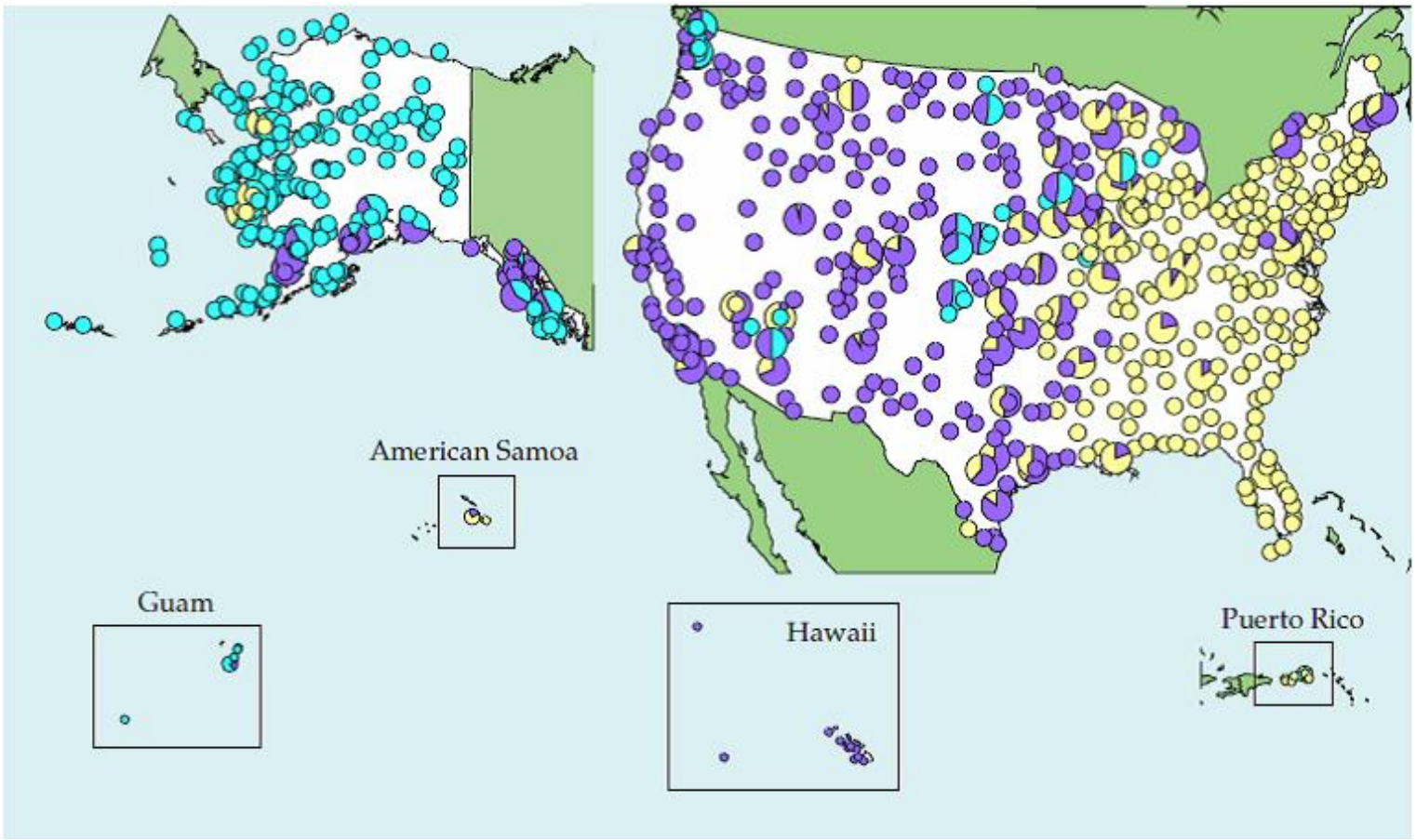

Figure: Overlapping communities in the network of US passenger air transportation. The three communities produced by the calculation correspond roughly to the east and west coasts of the country and Alaska.

Detecting non overlapping communities

- Supplement an extra step: assigning vertices to the community for which the value $\frac{k_{iz}}{k_z}$ is largest.

Results for running time

- We have 3 different algorithms:
  - Naive, $\delta = 0$: using naive EM algorithm.
  - Fast, $\delta = 0$: using pruned algorithm with $\delta = 0$.
  - Fast, $\delta = 0.001$: using pruned algorithm with $\delta = 0.001$

Results for running time

Running conditions	Time (s)	Iterations	Log-likelihood
US air transportation, $n = 709$, $m = 3327$, $K = 3$			
naive, $\delta = 0$	15.71	55719	−8924.58
fast, $\delta = 0$	14.67	55719	−8924.58
fast, $\delta = 0.001$	2.17	26063	−9074.21
Network science collaborations [43], $n = 379$, $m = 914$, $K = 3$			
naive, $\delta = 0$	0.93	13165	−3564.74
fast, $\delta = 0$	0.82	13165	−3564.74
fast, $\delta = 0.001$	0.13	10747	−3577.85
Network science collaborations, $n = 379$, $m = 914$, $K = 10$			
naive, $\delta = 0$	3.19	18246	−2602.15
fast, $\delta = 0$	3.15	18246	−2602.15
fast, $\delta = 0.001$	0.49	12933	−2611.96
Network science collaborations, $n = 379$, $m = 914$, $K = 20$			
naive, $\delta = 0$	6.16	19821	−2046.95
fast, $\delta = 0$	6.09	19821	−2046.95
fast, $\delta = 0.001$	0.94	14010	−2094.85

Running conditions	Time (s)	Iterations	Log-likelihood
Political blogs [44], $n = 1490$, $m = 16\,778$, $K = 2$			
naive, $\delta = 0$	11.42	13773	−48761.1
fast, $\delta = 0$	11.46	13773	−48761.1
fast, $\delta = 0.001$	4.14	13861	−48765.6
Physics collaborations [45], $n = 40\,421$, $m = 175\,693$, $K = 2$			
naive, $\delta = 0$	4339.57	424077	-1.367×10^6
fast, $\delta = 0$	2557.91	424077	-1.367×10^6
fast, $\delta = 0.001$	253.41	61665	-1.378×10^6
Amazon copurchasing [46], $n = 403\,394$, $m = 2\,443\,408$, $K = 2$			
naive, $\delta = 0$	170646.9	1222937	-2.521×10^7
fast, $\delta = 0$	105042.3	1222937	-2.521×10^7
fast, $\delta = 0.001$	11635.0	120612	-2.538×10^7
LiveJournal [47, 48], $n = 4\,847\,571$, $m = 42\,851\,237$, $K = 2$			
fast, $\delta = 0$	39834.3	26163	-4.611×10^8
fast, $\delta = 0.001$	3093.7	1389	-4.660×10^8

Conclusion

- Advantages
  - detects on both overlapping and non-overlapping communities.
  - has a rigorous mathematical foundation.
  - can apply to networks of millions of vertices.
  - gives results competitive with privious algorithms.
- Disadvantages
  - offers no criterion for determining K.